

Testen und Analysieren von Software

Fehlerbehebung und Re-Engineering

WS 2002/2003



Univ.-Prof. em. Dr. H.-J. Hoffmann/
Dipl.-Inform. Ludger Martin

(Stand 10. Dezember 2002)

Die Literaturangaben dienen als
Anleitung zur jeweiligen Thematik.
Die Bearbeiter sollten sich um einen
darüber hinausgehenden Überblick
bemühen.

Angaben noch unvollständig !



- W.W. Royce: Managing the development of large software systems, concepts and techniques; Proc. WESCON, August 1979 (auch Proc. 9th Intl. Conf. Software Engineering, 1987, 328 - 338)
- J. Rumbaugh: Over the waterfall and into the whirlpool; Journal Object-Oriented Programming, 5 (1992) 2, 23 - 26
- B.W. Boehm: A spiral model of software development and enhancement; IEEE Computer, 21 (1988) 5, 61 - 72
- B. Boehm et al.: Using the WinWin spiral model, a case study; IEEE Computer 31 (1998) 7, 33 - 44
- V.T. Rajlich, K.H. Bennett: A staged model for the software life cycle; IEEE Computer, 33 (2000) 7, 66 - 71





- D. Graham: Requirements and testing - seven missing-link myths;
IEEE Software, 19 (2002) 5, 15 - 17
- J. Moore: ISO 12207 and related software life-cycle standards;
<http://www.acm.org/tsc/lifecycle.html> (Zugriff
1. Okt. 2002)
- L. Cray: ISO/IEC 12207 Software lifecycle processes
<http://www.stsc.hill.af.mil/crosstalk/1996/aug/isoiec.asp> (Zugriff
1. Okt. 2002)

#02
18.11.02

Was sind "Fehler"? (I)

Qualitätsbegriff (auch bei WWW)



- H. Lieberman: The debugging scandal and what to do about it; Comm. ACM, 40 (1997), 4, 26 - 29 (und weitere Aufsätze in diesem Heft)
- L. Prechelt: Begriffe; 1999
<http://www.ipd.uka.de/~prechelt/swt2/node33.html> (Zugriff 2. Oktober 2002)
- B.Kahlbrandt: ... Qualitätsbegriff bei Software; 2001
<http://www.informatik.fh-hamburg.de/~sep/seq11/node32.html> (Zugriff 2. Okt. 2002)
- P.C. Jorgensen: Software testing, a craftsman's approach; CRC Press, 2002
- NN: Building bug-free O-O software — an introduction to *Design by Contract*TM; 2002 (Zugriff 2. Oktober 2002)
<http://archive.eiffel.com/doc/manuals/technology/contract/page.html>
- B. Marick: Failure improvement; (Zugriff 2. Oktober 2002)
<http://www.testingcraft.com/failure-improvement.html>



#02
18.11.02

Was sind "Fehler"? (II)

Qualitätsbegriff (auch bei WWW)



- J. Offutt: Quality attributes of Web software applications; IEEE Software 19 (2002) 2, 25 - 31
- E. Heatt, R. Mee: Going faster, testing the Web application; IEEE Software 19 (2002) 2, 60 - 65
- NN: Ihre Rechte beim Computerkauf; 1998
<http://hosting-service.euregio.net/vsz/comput1f.htm> (Zugriff 2. Oktober 2002)
- J. Nielsen: Error message guidelines; Alertbox, June 24, 2001
<http://www.useit.com/alertbox/20010624.html> (Zugriff 4. Oktober 2002)

Zu den Referaten #03 - #06

(Allgemeines)



- A.V. Aho et al.: Compilerbau Teil 1 und 2; Oldenbourg, 1999
- U. Kastens: Übersetzerbau; Oldenbourg, 1990
- NN: Syntaxanalyse - Parser;
<http://www-ui.informatik.uni-oldenburg.de/ai/lehre/veranstaltungen/ws99/download/syntax1.pdf>, 1999 (Zugriff 2. Oktober 2002)
- NN: Types of errors; (Zugriff 10. Oktober 2002)
<http://gsbapp2.uchicago.edu/sas/sashtml/lrcon/z0993446.html>, 1999
- J. Elder: A Bibliography of Compiler Construction; 1994
[http://www.cs.qub.ac.uk/~D.McKeever/csc303/Reference/Bibliography.html#Lexical analysis](http://www.cs.qub.ac.uk/~D.McKeever/csc303/Reference/Bibliography.html#Lexical%20analysis), (Zugriff 2. Oktober 2002)
- P.N. van den Bosch: Bibliography on syntax error handling in context-free languages; ACM SIGPLAN Notices, 27 (1992) 4, 77 - 85
- W. Horwat: JavaScript 2.0 error recovery; 1999 (Zugriff 11. Okt. 2002)

#03
25.11.02

Lexikalische Fehler

(bei der Zeichenanalyse; Schreibfehler)



- I. Durham et al.: Spelling correction in user interfaces; Comm. ACM, 26 (1983) 10, 764 - 773
- M.A. Bickel: Automatic correction to misspelled names - a fourth-generation language approach; Comm. ACM, 30 (1987) 3, 224 - 228
- C. McManis: Lexical analysis and Java; 1997 (Zugriff 2. Oktober 2002)
http://www.javaworld.com/javaworld/jw-01-1997/jw-01-indepth_p.html
- K.A.S. Abdel-Ghaffar: Detecting substitutions and transposition of characters: The Computer Journal 41 (1998)4, 270 - 277

LEX bzw. *FLEX* (u.a.) als Entwicklungswerkzeuge für Übersetzerprogramme

#04
2.12.02

Syntaktische Fehler I

(bei zielgerichteter Analyse, "top-down")



- R. Völler: Top-Down-Analyse; (Zugriff 2. Oktober 2002)
<http://users.informatik.fh-hamburg.de/~voeller/fc/comp/node16.html>
- NN: LL parsing; (Zugriff 2. Oktober 2002)
<http://www.cs.duke.edu/courses/spring02/cps140/lects/sectllparseS.pdf>
- N. Wirth: Coping with syntax errors (Kapitel 7.3 in *Compiler Construction*); Addison-Wesley, 1996, 40 - 4
- J. Grosch: Efficient and comfortable error recovery in recursive descent parsers; *Structured Programming*, 11 (1990) 3, 129 -140
- S.O. Anderson et al.: An assessment of locally least-cost error recovery; *The Computer Journal*, 26 (1983) 1, 15 - 24

#05
9.12.02

Syntaktische Fehler II

(bei rückschauender Analyse, "bottom-up")



- NN: Problem bei SR-Parsern ...;
http://ui.informatik.uni-oldenburg.de/ai/lehre/veranstaltungen/ws99/download/LR_Parser.pdf, 1999 (Zugriff 9. Oktober 2002)
- R. Völler: Bottom-up-analysis; (Zugriff 2. Oktober 2002)
<http://users.informatik.fh-hamburg.de/~voeller/fc/comp/node17.html>
- S.L. Graham, S.P. Rhodes: Practical syntactic error recovery; ACM Communications, 18 (1975) 11, 639 - 650
- S. Sippu, J. Soisalon-Soininen: Practical error recovery in LR parsing; Principles of Programming Languages, 9 (1982), 177 - ???
- I.-S. Kim, K.-M. Choe: Error repair with validation in LR-based parsing; ACM Transact. Programming Languages & Systems, 23 (2001) 4, 451 - 471
- C. Cerecke: Repairing syntax errors in LR-based parsers; 2002
<http://www.cosc.canterbury.ac.nz/~cdc/pub/cerecke02a.pdf> (Zugriff 9. Okt. 2002)

YACC bzw. *BISON* (u.a.) als Entwicklungswerkzeuge für Übersetzerprogramme, ebenso *GNU CC*

“Semantische” Fehler (I)



- F. Hernandez-Campos: Semantic analysis; 2002
<http://www.cs.unc.edu/~fhernand/144/lectures/lect12.pdf> (Zugr. 11. Okt. 2002)
- P. Graham: Introduction to compilers - Semantic analysis notes;
<http://www.cs.umanitoba.ca/~cs329/Notes/4Seman.pdf>, 2002
(11. Okt. 2002)
- S.A. Edwards: Types and static semantic analysis; 2002 (11.10. 02)
<http://www.cs.columbia.edu/~sedwards/classes/2002/w4115/types.9up.pdf>
- C.W.Johnson, C. Runciman: Semantic errors - diagnosis and repair; ACM SIGPLAN Notices, 17 (1982) 6, 88 - 97
- F. Tip, T.B. Dinesh: A slicing-based approach for locating type errors; ACM Transact. Software Engineering Methodology, 10 (2001) 1, 5 - 55 (hier 5 - 16)

#06
16.12.02

“Semantische” Fehler (II)



- A. Boccalatte, M. Di Manzo: An approach to the detection of semantic errors; BCS Computer Journal, 23 (1980) 4, 317 - 323
- A. Boccalatte et al.: Error recovery with attribute grammars; BCS Computer Journal, 25 (1982) 3, 231 - 337
- G. Adorni et al.: Top-down semantic analysis; BCS Computer Journal, 27 (1984) 3, 233 - 237

Fehler bei interaktivem Rechnergebrauch

(I)



- M.Y. Ivory, M.A. Hearst: The state of the art in automating usability evaluation of user interfaces; ACM Computing Surveys, 33 (2001) 4, 470 - 516
- C.E. Wilson, K.P. Covne: Tracking usability issues - to bug or not to bug?; interactions, 8 (2001) 3, 15 - 19
- A. Parush: Usability design and testing; interactions, 8 (2001) 5, 13 - 17
- T.T. Hewett: Importance of failure analysis for human-computer interface design; Interacting with Computers 3 (1991) 1, 3 - ???
- T. Ostrand et al.: A visual test development environment for GUI systems; ACM Software Engineering Notes, 23 (1998) 2, 82 - 92
- D. Frohlich et al.: The management of repair in human computer action; <http://www-uk.hpl.hp.com/techreports/93/HPL-93-48.html> (nur Zusammenfassung), 1993
- H.E. Blanchard: ... human computer interaction standards ...; Regular column in ACM SIGCHI Bulletin
<http://www.acm.org/sigchi/bulletin/> (Zugriff 11. Oktober 2002)

Fehler bei interaktivem Rechnergebrauch

(II)



- A. M. Memon: GUI testing - pitfalls and process: IEEE Computer, 35 (2002) 8, 87 - 88
- E. Hieatt, R. Mes: Going faster - testing the Web application; IEEE Software, 19 (2002) 2, 60 - 65
- L. Upchurch et al.: Using card sorts to elicit Web page quality attributes; IEEE Software, 18 (2002) 4, 84 - 89
- W. Schweibenz: Heuristische Evaluation von Web-sites. In H. Oberquelle et al. (Herausg.): Mensch & Computer 2001; Teubner, 2001, 391 - 392
- J. Nielsen: Why you only need to test with 5 users; Alertbox, March 19, 2000, <http://www.useit.com/alertbox/20000319.html>, (Zugriff 2000)
- G. Versteegen: *Test the Web* - die Bedeutung von Softwaretests im Zeitalter des Internets; sw development, März/April 2001, 20 - 24
- Mehrere Autoren: Seminar *Webanalyse*; FG PÜ, TUD, WS 2001/02, <http://www.informatik.tu-darmstadt.de/PU/webanalyse/>, (Zugriff 2002)

#09

20.01.03

Terminänderung!

Analyse von Software - statisch - Betreuung L. Martin



- R. Tischler et al.: Static analysis of programs as an aid to debugging, ACM SIGPLAN Notices, 18 (1983) 8 (bzw. ACM Software Engineering Notes, 8 (1983) 4), 155 - 158
- H. Sneed: Statische Analyse objektorientierter Software (Prüfung, Messung und Entwurfsabgleichung); Softwaretechnik-Trends, 20 (2000) 2, 24 - ???
- S. Bassil, R.K. Keller: Software visualization tools - survey and analysis; 9th Intl. Workshop Program Comprehension, 2001
- H.A. Müller, K. Klashinsky: Rigi - A system for programming-in-the-large; Proc. 10th Internatl. Conf. Software Engineering, 1988, 80 -86

Analyse von Software - dynamisch - Betreuung L. Martin



- T. Souder et al.: Form - A framework for creating views of program executions; Proc. IEEE Intl. Conf. Software Maintenance, 2001, 612 - 620
- T. Systä: On the relationship between static and dynamic models in reverse engineering Java software; Proc. 6th Working Conf. Reverse Engineering, 1999, 304 - 313 (auch <http://www.cs.tut.fi/~tsysta/papers/ssta.ps.gz> (Zugriff 14. Oktober 2002))
- L. Peters: Introducing dynamic analysis into reengineering; <http://www.c3i.osd.mil/bpr/bprcd/5295.htm>, Sept. 1995 (10. Okt. 2002)
- L. Martin et al.: Dynamic component program visualization; Working Conf. Reverse Engineering, October 2002, 289 - 298

Fehlerabdeckung/ Regressionstest



- Y. Li, N.J. Wahl: An overview of regression testing; ACM Software Engineering Notes, 24 (1999) 1, 69 - 73
- A. McCarthy: Unit and regression testing; Dr. Dobbs Journal, 262 (1997) 2, 18 - 84, <http://www.ddj.com/articles/1997/9702/>, (10. Okt. 2002)
- E.J. Weyucker: The trouble with testing components. In G.T. Heineman, W. T. Councill: Component-based software engineering; Addison-Wesley, 2001, 499 - 512)
- L. Martin: Visual component integration and regression test; Workshop Component-based Software Development Processes, April 2002
- K.Beck: Simple Smalltalk testing - with patterns; Smalltalk report, 4 (1994) 2, 16 - 18 (siehe auch <http://www.xprogramming.com/testfram.htm> (Zugriff Febr. 2002))

“Re-Engineering” / “Re-Factoring” (I)

Betreuung L. Martin



- E.J. Chikofsky, J.H. Cross II: Reverse engineering and design recovery, a taxonomy; IEEE Software 7 (1990) 1, 13 - 17
- J. Bergey et al.: Reengineering strategies and options. In: Options Analysis for Reengineering - Issues and Conceptual Approach, <http://www.sei.cmu.edu/publications/documents/99.reports/99tn014/99tn014abstract.html>, Chapter 4, 2002 (Zugriff 10. Oktober 2002)
- H.A. Müller: Understanding software systems reverse engineering technologies research and practice; März 2001 (Zugriff 10. Okt. 2002)
<http://www.sei.cmu.edu/reengineering/guide/UVicRevTut/...x...>
?????????
X elem {*F3strat.html*, *F4rev.html*, *F5pu.html*, *F6tools.html*}
- C. Finkelstein: Business re-engineering and the internet: Transforming business for a connected world; (Zugriff 10. Okt. 2002)
<http://members.ozemail.com.au/~visible/papers/bripaper.htm>, März 2002





- D.R. Engler, W.C. Hsieh; DERIVE, a tool that automatically reverse-engineers instruction encoding; ACM SIGPLAN Notices 35 (2000) 7, 12 - 22
- P. Coffee: Re-factoring isn't a cure-all; April 2002
<http://www.eweek.com/article2/0.3959.9937.00.asp>, (Zugriff 10. Okt. 2002)
- K. Beck: Extreme Programming, das Manifest; Addison-Wesley, 2000
- D. Roberts et al.: A refactoring tool for Smalltalk; July 1997
<http://st-www.cs.uiuc.edu/~droberts/tapos.pdf> (Zugriff 10. Oktober 2002)
- J. Hoh: Refactoring; März 2002 (Zugriff 19. Oktober 2002)
http://ncstrl.informatik.uni-stuttgart.dee/ifi/ps/Lehre/HS_OO_Entwurf/Hoh.pdf

“Allgemeine Geschäftsbedingungen”



- Referat maximal 50 Minuten
- Anschließend Diskussion
 - inhaltlich
 - Vortragsstil
- Auf Anforderung mindestens fünf Tage früher kann PC und/oder Internet-Anschluß bereitgestellt werden
- Ausarbeitung maximal acht bis zehn Seiten DIN A4;
(in der Regel) nach einer Woche abzugeben (Papier);
Literaturangaben nicht vergessen!
- Nach Bestätigung (eventl. Korrekturwünsche) durch mich
Abgabe der endgültigen Ausarbeitung innerhalb einer
Woche als ppt-/doc-/pdf-Datei
(mit Namen “#xx-version-ii.ext”, “xx” elem {01 ÷ 11}, “ii” elem {1 ...})

